
Création et utilisation d'un résumé de métadonnées pour interroger efficacement des collections multimédias distribuées

Sébastien Laborie, Ana-Maria Manzat et Florence Sèdes

IRIT - Université Paul Sabatier

118 Route de Narbonne

31062 Toulouse Cedex 9

{Sebastien.Laborie, Ana-Maria.Manzat, Florence.Sedes}@irit.fr

RÉSUMÉ. Actuellement, de nombreux contenus multimédias sont créés à partir de plusieurs sources et stockés dans des environnements distribués. Pour éviter de centraliser l'ensemble des métadonnées d'un système et répondre efficacement à une requête d'un utilisateur, nous proposons d'engendrer et d'utiliser un résumé de métadonnées. Ce dernier aura pour fonction de localiser certaines unités de stockage qui contiennent les données multimédias désirées. L'originalité de ce résumé réside en le fait qu'il soit construit automatiquement sur la base des métadonnées extraites durant l'indexation. Dans cet article, nous montrons comment construire un tel résumé et illustrons notre approche au moyen de technologies issues du Web Sémantique, telles que RDF et SPARQL pour représenter et interroger des métadonnées sémantiques.

ABSTRACT. Currently, many multimedia contents are acquired and stored in real time and on different locations. In order to retrieve efficiently some desired information and avoid centralizing all metadata, we propose to compute a metadata summary, i.e., a concise version of the whole metadata, which guides the location of some desired multimedia contents. The originality of this summary is that it is automatically constructed based on the extracted metadata. In this paper, we show how to construct such summary and illustrate our framework with current Semantic Web technologies, such as RDF and SPARQL for representing and querying semantic metadata.

MOTS-CLÉS : Indexation et recherche de contenus multimédias distribués, résumé de métadonnées, RDF et SPARQL.

KEYWORDS: Distributed multimedia indexing and retrieval, Metadata summarization, RDF and SPARQL.

1. Introduction

Actuellement, de nombreuses activités humaines engendrent des contenus multimédias qui peuvent être capturés en temps réel et stockés dans des environnements éventuellement hétérogènes, éloignés géographiquement, de capacités diverses... Ceci est notamment le cas lors de l'utilisation de plates-formes mobiles, comme les téléphones portables ou bien les assistants personnels, mais également pour les domaines qui concernent la vidéo surveillance, les données médicales, etc.

Dans le même temps, l'essor de ces multiples contenus multimédias s'est accompagné d'un accroissement des métadonnées, c'est-à-dire des données qui décrivent ces contenus. Dans ce contexte d'abondance d'informations, il apparaît alors nécessaire de pouvoir répondre de manière rapide à une requête d'un utilisateur.

Pour éviter de centraliser la totalité des métadonnées d'un système, (Hinds *et al.*, 1998; Pietarila *et al.*, 2005; Roth *et al.*, 2006) prônent l'utilisation d'architectures distribuées dans lesquelles les contenus multimédias ainsi que leurs métadonnées sont stockés sur de multiples serveurs. Par conséquent, ces serveurs sont capables de retrouver en parallèle une information désirée, ce qui permet de répondre efficacement à une requête. Cependant, dans ces approches tous les serveurs sont interrogés alors que seule une partie d'entre eux contient l'information souhaitée par l'utilisateur. De plus, avec l'avènement d'un Web Sémantique (Berners-Lee *et al.*, 2001) dans lequel les métadonnées sont de plus en plus expressives, interroger de telles descriptions sur chaque serveur peut rapidement devenir coûteux en terme de temps d'exécution d'une requête.

Au sein de ce cadre d'informations distribuées, afin de retrouver efficacement des contenus multimédias, nous proposons d'engendrer et d'utiliser un résumé de métadonnées. Ce résumé est construit automatiquement sur la base des métadonnées extraites durant l'indexation et aura pour fonction de localiser certains serveurs qui contiennent les données multimédias désirées. En effet, certains éléments provenant des descriptions des contenus multimédias permettent de caractériser de façon générale l'information stockée sur un serveur. En simplifiant la requête initiale d'un utilisateur en vue d'interroger ces informations générales présentes dans le résumé, il est ensuite possible d'interroger uniquement certains serveurs qui peuvent potentiellement répondre à la requête initiale.

Tout d'abord, nous décrivons dans la section 2 un exemple d'application qui illustre l'utilité d'un résumé de métadonnées et présentons différents systèmes d'information distribuée manipulant des contenus multimédias. Dans la section 3, nous proposons une architecture générique d'indexation et de recherche de contenus multimédias distribués qui utilise un résumé de métadonnées. Dans cette partie, nous montrons tout particulièrement comment construire un résumé de métadonnées qui soit basé sur des descriptions RDF (Resource Description Framework) (Manola *et al.*, 2004). Ensuite, nous détaillons dans la section 4 le processus de localisation et de sélection de serveurs à partir de requêtes spécifiées dans le langage SPARQL (Prud'hommeaux *et*

al., 2008). Enfin, nous présentons dans la section 5 des résultats expérimentaux qui démontrent l'utilité ainsi que l'efficacité d'un résumé de métadonnées.

2. Indexer et rechercher des contenus multimédias distribués

2.1. Un exemple de système de gestion de contenus multimédias distribués

Considérons un système d'information dans lequel les contenus multimédias sont capturés par une flotte de véhicules de police, tels que des vidéos et des images (e.g., capturées à l'aide de caméras), des contenus sonores (e.g., les discussions radio) ainsi que des données textuelles (e.g., les rapports de policiers). Comme illustré dans la figure 1, les véhicules sont associés à de multiples serveurs dans lesquels les contenus multimédias sont stockés.

Pour retrouver une information particulière, comme par exemple *"Trouvez des vidéos qui contiennent une voiture bleue dans Paris avec le numéro d'immatriculation 1234 AB 56"*, un serveur central a connaissance de l'ensemble des serveurs distants qui contiennent les contenus souhaités et peut ainsi répondre à ce type de requête.

Une méthode naïve permettant de retrouver l'information souhaitée pourrait consister à stocker sur le serveur central l'ensemble des métadonnées du système d'information et y exécuter la requête de l'utilisateur.

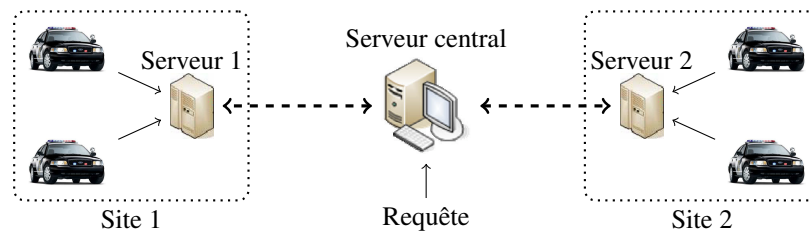


Figure 1 – Un exemple de système d'information distribuée.

Cependant, cette stratégie a de nombreux inconvénients :

- **Lenteur de traitement d'une requête** : Traiter une requête sur la totalité des métadonnées du système a de fortes chances de surcharger le serveur central, surtout lors du traitement de requêtes complexes et lorsque plusieurs requêtes sont traitées simultanément.

- **Surcharge de la bande passante** : Tous les contenus multimédias ou bien toutes les métadonnées doivent être transférés par le réseau au serveur central.

- **Centralisation du système** : Si le serveur central ne répond plus, l'ensemble des métadonnées doit de nouveau être recalculé et renvoyé sur un serveur central. De plus, dans le cas d'un système d'information dynamique mettre à jour le serveur central serait très coûteux.

– **Non-respect des droits sur les données** : Certaines métadonnées ne doivent pas être stockées sur le serveur central pour des raisons, par exemple, de vie privée.

Pour éviter tous ces inconvénients, nous pouvons tirer parti du contexte distribué en indexant et stockant localement les contenus multimédias et leurs métadonnées. Pour répondre rapidement à une requête, nous proposons de transférer sur le serveur central uniquement une version concise des métadonnées, que nous appelons dans la suite “*résumé*”. Par conséquent, le serveur central peut répondre directement à des requêtes très générales ou bien, pour des requêtes plus détaillées, localiser les serveurs distants qui peuvent potentiellement contenir des résultats. En ce qui concerne ces dernières, lorsqu’un ensemble de serveurs a été identifié, chacun d’entre eux peut traiter la requête de manière indépendante.

Ainsi, le transfert de données par le réseau est minimisé, car seulement une partie des métadonnées est envoyée sur le serveur central. De plus, les requêtes détaillées d’un utilisateur ne sont transférées et traitées simultanément que sur certains serveurs, ce qui allège le serveur central.

Dans la partie suivante, nous comparons notre approche qui utilise un résumé de métadonnées avec d’autres approches existantes.

2.2. *Approches existantes*

Ces dernières années, de nombreux travaux de recherche ont été conduits dans le domaine de l’indexation et de la recherche de contenus multimédias. De nombreux projets existants se focalisent sur certains types de contenus multimédias, comme par exemple exclusivement des images, alors que d’autres s’intéressent à tous types de médias. Une autre différence entre ces approches est le processus d’indexation et plus particulièrement le format des métadonnées qui peut être non sémantiquement défini comme les vecteurs de mots-clés, ou bien sémantiquement défini. Enfin, certains travaux considèrent une centralisation des métadonnées alors que d’autres les distribuent. Dans ce qui suit, nous proposons quelques approches récentes.

Un système centralisé pour retrouver tous types de contenus multimédias est présenté dans (Ayache *et al.*, 2006). Celui-ci utilise les machines à vecteurs de support (SVM) pour classer et retrouver les contenus multimédias. Les métadonnées sont construites à l’aide de descripteurs de bas niveau ainsi que de données sur le contexte obtenu à partir de contenus annotés dans le cadre de TRECVID¹.

Le projet CANDELA (Pietarila *et al.*, 2005) se consacre exclusivement autour de l’indexation et de la recherche de contenus vidéos. Le projet se concentre notamment sur le format standard de vidéo MPEG-7 obtenu à partir de l’indexation et propose de stocker ces descriptions dans des bases de données distribuées. Cette opération de stockage est néanmoins limitée en ce qui concerne les métadonnées sémantiques.

1. <http://www-nlpir.nist.gov/projects/trecvid/>

Le projet CAIM² se concentre sur la gestion d'images dans des environnements mobiles distribués. Les contenus multimédias sont capturés par des téléphones portables et envoyés sur un serveur central qui identifie les informations sur ces contenus.

Un moteur de recherche de contenus distribués basé un système multi-agents est proposé dans (Roth *et al.*, 2006). Dans un tel système, les contenus multimédias sont distribués sur des serveurs et indexés par des agents qui migrent d'un serveur à un autre. Pour faire face aux multiples métadonnées réparties, (Roth *et al.*, 2006) proposent deux types d'architectures : une dans laquelle les métadonnées sont centralisées et l'autre dans laquelle les métadonnées sont distribuées. Cependant, dans le second cas le serveur central n'est pas utilisé.

Les travaux de recherche les plus proches de notre approche sont ceux décrits dans (Hinds *et al.*, 1998). Ceux-ci proposent une version améliorée du système de la NASA d'observation de la Terre. Dans cette version les métadonnées sont distribuées et (Hinds *et al.*, 1998) proposent de stocker sur le serveur central une description hiérarchique d'un ensemble d'informations qui localise les serveurs contenant une information désirée. La principale différence avec notre approche est que cette description est a priori élaborée sur le serveur central.

Dans la section suivante, nous montrons comment construire de manière automatique un résumé de métadonnées basé sur des descriptions extraites lors du processus d'indexation.

3. Une architecture générique d'indexation de contenus multimédias distribués

Une architecture générique d'indexation de contenus multimédias distribués est proposée dans la figure 2. Celle-ci est basée sur l'exemple présenté dans la section 2.1 qui comprend un serveur central et plusieurs serveurs distants composés de contenus multimédias ainsi que de leurs métadonnées.

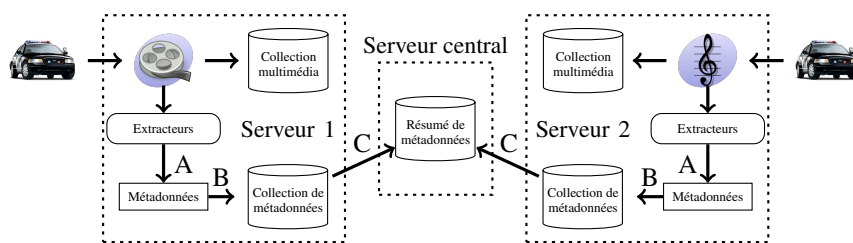


Figure 2 – Une architecture générique d'indexation de contenus multimédias distribués.

2. <http://caim.uib.no/>

Lorsqu'un contenu multimédia est capturé par un véhicule de police, celui-ci est stocké dans la collection multimédia et indexé par de multiples extracteurs. Ces extracteurs produisent les métadonnées correspondantes, par exemple, dans le cas d'une vidéo capturée par un véhicule de police, les extracteurs de contenu peuvent identifier des véhicules, leurs plaques d'immatriculation, etc. L'ensemble des métadonnées est ensuite stocké dans la collection de métadonnées. Enfin, une partie des collections de métadonnées, nommé "*résumé de métadonnées*", est transféré sur le serveur central.

Nous proposons de décrire plus en détail chaque élément de cette architecture :

- Un **contenu multimédia** fait référence à un type de média, tel que du texte, de l'image, une vidéo ou un contenu sonore.

- Une **collection multimédia** contient plusieurs contenus multimédias. Chaque collection multimédia est stockée sur un serveur dédié, comme par exemple un serveur correspondant à une catégorie de véhicules de police.

- Un ensemble d'**extracteurs** appliqué sur un contenu multimédia identifie un ensemble de métadonnées correspondantes. Des exemples d'extracteurs ont été proposés dans (Lambolez *et al.*, 1995) pour le texte, (Ma *et al.*, 1999) pour les images, (Kankanhalli *et al.*, 2000) pour les vidéos et (Gauvain *et al.*, 2001) pour les contenus sonores.

- Une **métadonnée de contenu** contient des informations sur les caractéristiques d'un média (e.g., taille, nom du fichier, type du fichier) ainsi que sur son contenu, comme par exemple une identification de la présence d'une personne ou bien une reconnaissance de numéros d'une plaque d'immatriculation d'un véhicule.

- Une **collection de métadonnées** contient l'ensemble des métadonnées de contenus qui décrivent une collection multimédia. Les métadonnées peuvent être encodées à l'aide de plusieurs standards, tels que les formats Exif, Dublin Core, MXF, etc. De plus, une collection de métadonnées fait référence aux contenus stockés dans la collection multimédia qui lui est associée.

- Un **résumé de métadonnées** est une version concise des collections de métadonnées. Le résumé est utile pour localiser un ensemble de serveurs qui contient une information particulière. Par exemple, le résumé peut identifier un ensemble de serveurs qui contient des données multimédias présentant des véhicules sans donner plus de détails sur ceux-ci.

Grâce à cette architecture générique, plusieurs moteurs d'indexation peuvent être intégrés sur chaque serveur. Plus précisément, les extracteurs de contenu peuvent correspondre à de multiples moteurs d'indexation. De plus, des moteurs d'indexation différents peuvent être intégrés sur chaque serveur distant.

Dans les parties suivantes, nous proposons de décrire en détail les processus A, B et C mis en évidence dans la figure 2.

3.1. Application des extracteurs sur des contenus multimédias (étape A)

Un extracteur appliqué sur un contenu multimédia identifie des informations qui concernent ce contenu, telles que l'occurrence de personnes, de véhicules, de sujets de discussions, etc. Par exemple, un système de reconnaissance de numéros de plaques d'immatriculation correspond à un certain type d'extracteur.

Les extracteurs sont des composants importants d'un système d'information car ce sont eux qui identifient les propriétés d'un contenu multimédia. En outre, lorsque de nombreux extracteurs sont appliqués sur un même contenu multimédia, cela enrichit d'autant plus sa description, ou plus précisément ses métadonnées correspondantes.

Nous proposons d'illustrer dans l'exemple suivant comment des descriptions RDF peuvent être combinées lorsque plusieurs extracteurs sont appliqués sur un même contenu multimédia.

Exemple 1 *RDF (Resource Description Framework) (Manola et al., 2004) est un langage qui permet de décrire des ressources. Une description RDF peut être représentée à l'aide d'un graphe orienté étiqueté comme illustré dans la figure 3*

Soit C_1 un contenu vidéo. Nous associons à C_1 une URI qui identifie de manière unique cette ressource, e.g., $ex:C_1$ où ex : fait référence à un domaine particulier. Ensuite, plusieurs extracteurs sont appliqués au contenu C_1 : un extracteur qui identifie des personnes avec leur couleur de cheveux (E_1) et un extracteur qui identifie des véhicules ainsi que certaines caractéristiques (E_2). Dans cet exemple, nous supposons que les données de sortie de chaque extracteur sont des descriptions RDF ou bien que celles-ci sont traduites en descriptions RDF.

La figure 3 présente l'ensemble des descriptions RDF extraites sur le contenu vidéo C_1 . Comme le montre cette figure, certains extracteurs peuvent être composés de plusieurs extracteurs. C'est le cas notamment de E_2 qui est composé de deux extracteurs : un qui identifie les numéros de plaque d'immatriculation (E_{2a}) ainsi qu'un autre qui identifie la couleur des véhicules (E_{2b}).

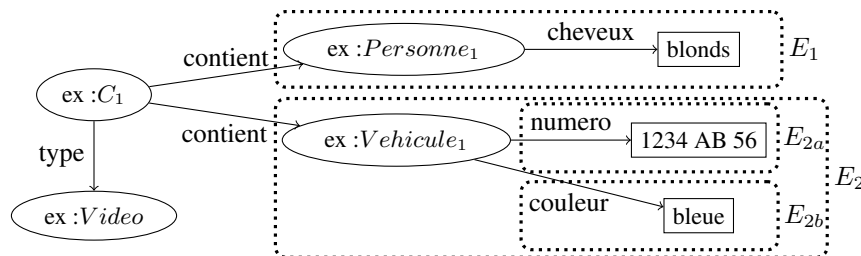


Figure 3 – Une description RDF du contenu multimédia C_1 .

Lorsque plusieurs extracteurs sont appliqués sur un contenu multimédia, les métadonnées extraites sont agrégées (ou liées) à l'URI qui correspond à ce contenu multimédia. Par conséquent, après application d'un ensemble d'extracteurs, la description de la figure 3 signifie que C_1 est une vidéo qui contient une personne blonde ainsi qu'un véhicule bleu identifié par le numéro de plaque d'immatriculation 1234 AB 56.

Comme illustré dans l'exemple précédent, la construction des métadonnées de contenu, notamment pour des descriptions RDF, peut être complètement automatisée.

3.2. Construire des collections de métadonnées (étape B)

Une collection de métadonnées contient plusieurs métadonnées de contenu. Elle peut également contenir des informations supplémentaires qui concernent le serveur mais aussi les métadonnées de contenu extraites (comme par exemple, des informations qui n'ont pas été extraites par les extracteurs). Ces informations additionnelles sont nommées dans la littérature comme la notion de contexte. (Mostefaoui *et al.*, 2004) présente différentes définitions de cette notion de contexte mais aussi propose des méthodes pour l'identifier et le modéliser.

Ces informations additionnelles peuvent être ajoutées manuellement à la collection de métadonnées ou bien être déduites à l'aide de règles d'inférence qui ont été spécifiées a priori. Il est important de souligner qu'avec ou sans ces informations supplémentaires, les utilisateurs du système d'information peuvent néanmoins retrouver des informations sur les contenus multimédias qui ont été extraites durant le processus d'indexation.

L'exemple qui suit montre comment construire une collection de métadonnées et lui attacher des informations de contexte.

Exemple 2 Soient $Serveur_1$ une collection de métadonnées d'un serveur, et C_1 et C_2 deux contenus multimédias. Nous associons à $Serveur_1$ une URI qui identifie de manière unique $Serveur_1$ et agrégeons à cette URI l'ensemble des descriptions RDF produites par les extracteurs, c'est-à-dire l'ensemble des métadonnées de contenus. La figure 4 illustre le contenu de cette collection de métadonnées décrite en RDF.

Dans cette figure, des informations supplémentaires au sujet du contexte ont été ajoutées, telle que la localisation du $Serveur_1$ (mis en évidence en bleu dans la figure 4). De plus, grâce à l'identification du numéro de plaque d'immatriculation du véhicule nommé $Vehicule_1$, des informations qui le concernent ont aussi pu être déduites et ajoutées, comme par exemple le nom du propriétaire du véhicule (mis en évidence en rouge dans la figure 4).

Ce type d'informations additionnelles, qui concernent le contexte, peut être spécifié à l'aide d'ontologies, à l'instar de (Strang et al., 2003). Notons que ces derniers ont aussi développé un moteur d'inférence qui engendre ces informations additionnelles.

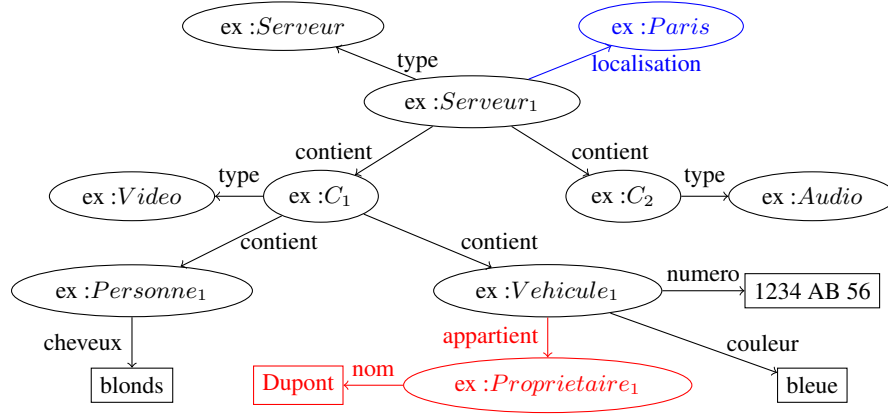


Figure 4 – Une description RDF du contenu de la collection de métadonnées du *Serveur₁*.

3.3. Construire un résumé de métadonnées (étape C)

Dans un objectif de répondre efficacement à une requête d'un utilisateur en localisant les serveurs qui contiennent une information particulière, nous proposons de transférer sur le serveur central uniquement une version concise des collections de métadonnées. Si l'on se concentre sur la collection de métadonnées illustrées dans la figure 4, il est possible de résumer cette description en ne retenant que le fait que ce serveur localisé à Paris regroupe un contenu sonore et une vidéo qui présente des personnes et des véhicules (sans donner plus de détails sur ces personnes ou ces véhicules). Une version encore plus concise pourrait consister à dire que ce serveur localisé à Paris est composé d'un contenu vidéo et sonore (sans donner plus de détails sur les contenus).

Par conséquent, à partir d'une collection de métadonnées, il est possible d'engendrer de multiples résumés. De plus, ces résumés sont des parties de collection qui peuvent varier en fonction de la précision que l'on souhaite lui donner. En vue de sélectionner des parties de collection de métadonnées, nous proposons de définir la notion de degré de résumé, ou bien de granularité d'un résumé, telle que :

- $\text{degré} = 0$ signifie qu'aucune information n'est sélectionnée.
- $0 < \text{degré} < n$ signifie qu'une partie de l'information contenue dans une collection de métadonnées est sélectionnée. L'importance de cette sélection dépend de la valeur du degré. Dans notre cas, plus le degré sera faible, plus le résumé sera concis.
- $\text{degré} = n$ signifie que l'ensemble de la collection de métadonnées est sélectionné, c'est-à-dire que l'on préserve l'ensemble des informations de la collection.

Nous proposons dans l'exemple suivant de montrer l'utilité de cette notion de degré sur des descriptions RDF et présentons une méthode qui permet de résumer de telles descriptions.

Exemple 3 *Considérons la description RDF illustrée dans la figure 4. Pour résumer cette description, nous devons nous focaliser sur l'URI du Serveur_1 et sélectionner les informations environnantes qui lui sont liées. En effet, l'URI du Serveur_1 est le principal sujet de notre résumé, sélectionner ses informations environnantes ne fera qu'accroître sa description. De plus, à l'aide de la représentation sous forme de graphe d'une description RDF, il est possible de constater que plus l'information est éloignée de l'URI du serveur, plus on dispose de détails, non seulement sur les données du serveur lui-même, mais aussi sur les descriptions qui lui sont liées.*

À partir de cette constatation, nous proposons de sélectionner à partir de l'URI du serveur toutes les métadonnées présentes sur un chemin d'une certaine longueur. Dans ce cadre, la longueur de ce chemin correspondra à la notion de degré du résumé. Par conséquent, lorsque l'on applique cette méthode sur la description RDF de la figure 4 :

- Si $\text{degré} = 0$, le résumé contient seulement l'URI du Serveur_1 , en outre aucune information sur ce serveur.
- Si $0 < \text{degré} < 4$, le résumé contient l'URI du Serveur_1 ainsi qu'un sous-graphe de la figure 4. Par exemple, si $\text{degré} = 2$, le résumé spécifiera que le Serveur_1 contient deux données multimédias : un contenu sonore et une vidéo dans laquelle des personnes ainsi que des véhicules ont été identifiés (cf., figure 5).
- Si $\text{degré} = 4$, le résumé correspond strictement à la description de la figure 4, c'est-à-dire qu'elle contient l'ensemble des informations de la collection de métadonnées.

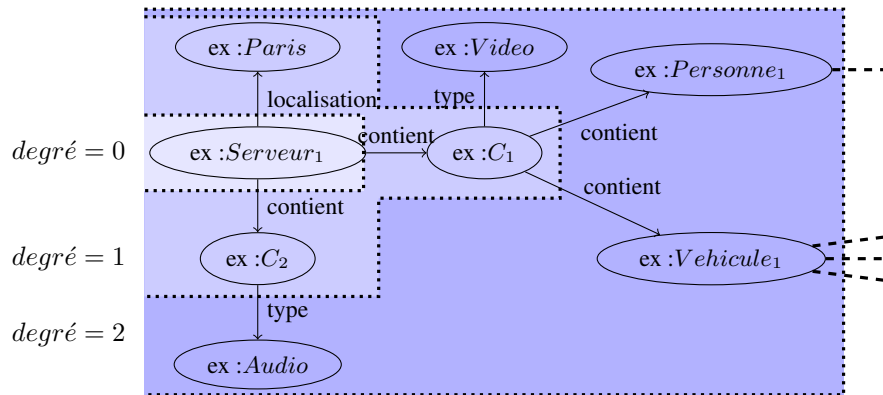


Figure 5 – Une description RDF résumée de la figure 4 avec $0 \leq \text{degré} \leq 2$.

Cette méthode pour construire des résumés peut être complètement automatisée et dépend uniquement du contenu des collections de métadonnées. Par conséquent, cette approche est indépendante du contexte d'application des systèmes d'information. De plus, cette approche peut être assistée manuellement en ajoutant des éléments considérés comme importants ou en supprimant des éléments moins importants. Enfin, différents degrés de résumés peuvent être appliqués sur les collections de métadonnées.

Chaque résumé engendré sur chaque serveur sera transféré et fusionné sur le serveur central³. Nous montrons dans la section suivante comment localiser certains serveurs à partir du résumé et d'une requête d'un utilisateur.

4. Interrogation du résumé et des collections distribuées de métadonnées

Comme nous l'avons montré dans la figure 2 et décrit dans les parties précédentes, les collections de métadonnées sont distribuées et un résumé de métadonnées est engendré sur le serveur central. Celui-ci est utilisé pour répondre à des requêtes très générales d'un utilisateur ou bien, pour des requêtes plus détaillées, localiser les serveurs distants qui contiennent potentiellement l'information souhaitée.

Par exemple, supposons un utilisateur qui spécifie la requête q suivante "*Trouve des vidéos de véhicules et donne-moi leurs numéros de plaque d'immatriculation.*". Lorsque cette requête est exécutée sur le serveur central (qui contient le résumé de métadonnées), si le serveur peut répondre à cette requête, celui-ci envoie les résultats à l'utilisateur. Dans le cas contraire, c'est-à-dire lorsqu'aucun résultat n'a été trouvé, cela signifie probablement que certains serveurs distants peuvent contenir l'information désirée.

Pour éviter d'exécuter la requête sur l'ensemble des serveurs distants, nous proposons de transformer la requête q en q' , telle que q' cherche les serveurs qui contiennent l'information souhaitée dans la requête q . Dans l'exemple, q' sera "*Trouve les serveurs qui contiennent des vidéos de véhicules et donne-moi leurs numéros de plaque d'immatriculation.*". Cependant, vu que l'exécution de la requête q n'a pas eu de résultats à cause du résumé de métadonnées, l'exécution de la requête q' ne retrouvera pas non plus de résultats. Par exemple, l'exécution de q ou de q' sur le résumé de métadonnées illustré dans la figure 5 ne contient pas, dans les deux cas, d'éléments de réponse en ce qui concerne les numéros des plaques d'immatriculation.

Pour localiser les serveurs qui contiennent potentiellement une réponse, la requête q' doit être relaxée en éliminant certains de ses éléments. Par exemple, q' peut être simplifiée en q'' avec $q'' =$ "*Trouve les serveurs qui contiennent des vidéos de véhicules.*".

Lorsqu'un ensemble de serveur a été identifié, la requête initiale q peut être ensuite exécutée en parallèle sur chaque serveur distant pour retrouver l'information souhaitée

3. Une définition de la fusion de descriptions RDF est disponible à l'adresse suivante : <http://www.w3.org/TR/2001/WD-rdf-mt-20010925/#merging>.

par l'utilisateur. L'union de l'ensemble des résultats produits par ces serveurs sera présenté à l'utilisateur.

Nous proposons dans l'exemple suivant de montrer l'applicabilité d'une telle approche sur des requêtes spécifiées à l'aide du langage SPARQL.

Exemple 4 *SPARQL (Prud'hommeaux et al., 2008) est un langage de requête qui permet d'interroger des descriptions RDF. Considérons la requête q qui cherche des vidéos de véhicules ainsi que leurs numéros de plaques d'immatriculation, spécifiée en SPARQL :*

```
SELECT *
FROM <Resume.rdf>
WHERE {
    ?video <type> <ex :Video> .
    ?video <contient> ?vehicule .
    ?vehicule <numero> ?numero .
}
```

Supposons que cette requête renvoie 0 résultat sur le résumé de métadonnées, il est nécessaire de localiser certains serveurs distants contenant potentiellement une réponse à cette requête. Pour transformer la requête q en q' , nous ajoutons une variable ?serveur de type Serveur pour localiser ces serveurs distants, c'est-à-dire ?serveur <type> Serveur. Ensuite, deux cas peuvent se produire : la requête q cherche un contenu ou bien la requête cherche uniquement certaines métadonnées indépendamment d'un contenu. Dans cet article, nous considérerons uniquement le premier cas⁴.

Si la requête recherche un contenu multimédia, comme nous savons que chaque serveur "contient" des contenus multimédias, nous pouvons lier la variable ?serveur au contenu recherché identifié par la variable ?video, c'est-à-dire en ajoutant dans la requête q' l'information suivante : ?serveur <contient> ?video.

Pour retrouver des résultats sur le serveur central à partir de la requête q' , nous devons relaxer q' en q'' . De nombreux travaux de recherche ont été conduits pour simplifier des requêtes SPARQL, comme dans (Quilitz et al., 2008) et (Hurtado et al., 2008). Le but de cet article étant de promouvoir l'utilisation d'un résumé de métadonnées, nous ne détaillerons aucun algorithme de relaxation de requêtes. Cependant, il est possible de constater, qu'une solution envisageable pourrait consister à retirer certaines variables de la requête q' .

4. Pour le deuxième cas, comme nous ne savons pas comment les métadonnées recherchées sont liées à la variable ?serveur, il serait nécessaire d'utiliser dans la requête des expressions régulières comme définies dans PPARQL (Alkhateeb, 2008), i.e., ?serveur +(?property) ?aVariable. Cependant, ce type de formalisme n'est pas présent dans le standard SPARQL.

La section suivante présente des résultats expérimentaux qui démontrent l'efficacité d'un résumé de métadonnées.

5. Résultats Expérimentaux

Dans un premier temps, nous avons implémenté notre proposition en Java en utilisant JENA⁵ pour la gestion des descriptions RDF, et ARQ⁶ pour interroger ces descriptions à l'aide du langage SPARQL. Pour être le plus exhaustif possible, nous générons de façon aléatoire les descriptions RDF, en précisant leur taille ainsi que leur densité. Dans notre approche, les URI des contenus multimédias sont liés à chaque URI des serveurs distants qui leur sont associés (cf., section 3.2). Afin de préserver cette information, pour chaque description RDF engendrée de façon aléatoire, nous avons sélectionné au hasard un élément de celle-ci et lui avons attribué le type Serveur.

Par la suite, nous avons évalué l'efficacité du résumé de métadonnées en enregistrant le temps de réponse aux requêtes spécifiées par un utilisateur. Nous proposons d'exécuter :

- une requête initiale sur une collection centralisée de métadonnées qui fusionne l'ensemble des collections distantes n'ayant pas été résumé. Nous exécutons également cette même requête sur une seule collection distante.
- une requête relaxée sur la collection centralisée de métadonnées ainsi que sur un résumé de celle-ci.
- une requête relaxée qui identifie les serveurs susceptibles de contenir l'information recherchée. Cette requête est exécutée sur la collection centralisée résumé de métadonnées.

La figure 6 présente une requête SPARQL initiale (Requête 1), une requête relaxée (Requête 2) qui est obtenue de la Requête 1 en supprimant des éléments, et une requête relaxée qui cherche les serveurs distants contenant l'information recherchée (Requête 3). Nous avons choisi ces trois requêtes pour être le plus générique et exhaustif possible. La méthode utilisée pour relaxer la requête est très simple, le but étant de constater son influence sur le temps de réponse. Comprenant plus de variables, les trois autres requêtes présentées (Requêtes 4, 5 et 6) sont encore plus complexes.

Les deux tableaux de la figure 6 contiennent le temps de réponse, en ms, obtenu lors de l'exécution des requêtes sur différentes descriptions RDF. Tous les tests ont été effectués sur un système Mac OS X, Intel Core Duo CPU 2.16GHz avec 2GB RAM.

Dans le premier tableau, les résultats des expérimentations montrent que la taille des descriptions RDF a une forte influence sur le temps d'exécution de la requête. Par exemple, pour la Requête 1 on peut constater que le temps de réponse à cette requête sur une seule collection de métadonnées est inférieur au temps de réponse à la même

5. <http://jena.sourceforge.net/>

6. <http://jena.sourceforge.net/ARQ/>

<pre>SELECT * WHERE { ?s1 ?p1 ?o1 . ?o1 ?p2 ?o2 . ?o2 ?p3 ?o3 . }</pre>	<pre>SELECT * WHERE { ?s1 ?p1 ?o1 . ?o1 ?p2 ?o2 . }</pre>	<pre>SELECT ?s1 WHERE { ?s1 <type> Serveur . ?s1 ?p1 ?o1 . ?o1 ?p2 ?o2 . }</pre>
(a) Requête 1 : requête initiale.	(b) Requête 2 : Requête 1 relaxée.	(c) Requête 3 : Requête 1 relaxée qui cherche des serveurs.
<pre>SELECT * WHERE { ?s1 ?p1 ?o1 . ?o1 ?p2 ?o2 . ?o2 ?p3 ?o3 . ?o3 ?p4 ?o4 . }</pre>	<pre>SELECT * WHERE { ?s1 ?p1 ?o1 . ?o1 ?p2 ?o2 . ?o2 ?p3 ?o3 . }</pre>	<pre>SELECT ?s1 WHERE { ?s1 <type> Serveur . ?s1 ?p1 ?o1 . ?o1 ?p2 ?o2 . ?o2 ?p3 ?o3 . }</pre>
(d) Requête 4 : requête initiale.	(e) Requête 5 : Requête 4 relaxée.	(f) Requête 6 : Requête 4 relaxée qui cherche des serveurs.

nombre de serveurs	taille d'une collection distante	Requête 1		Requête 2		Requête 3
		résumé central degré=n	collection distante	résumé central degré=n	résumé central degré=3	résumé central degré=3
2	10	65	14	6	3	2
	20	284	90	20	6	4
	30	1030	453	62	11	4
	40	2960	1253	114	23	6
	50	6213	3099	216	45	7
5	10	96	11	10	5	4
	20	523	86	44	13	5
	30	2551	520	149	29	7
	40	6380	1243	284	37	9
	50	15663	3166	560	117	17

(g) Résultats de l'évaluation des requêtes 1, 2 et 3 en ms.

nombre de serveurs	taille d'une collection distante	Requête 4		Requête 5		Requête 6
		résumé central degré=n	collection distante	résumé central degré=n	résumé central degré=3	résumé central degré=3
2	10	150	47	15	4	3
	20	2037	886	183	7	3
	30	15083	7596	871	115	5
	40	54177	25573	2524	217	6
	50	170010	84516	6203	733	9
5	10	266	51	37	15	2
	20	5165	1160	412	61	4
	30	34653	7153	2057	188	7
	40	141310	27366	6372	922	13
	50	409200	80402	15265	1436	17

(h) Résultats de l'évaluation des requêtes 4, 5 et 6 en ms.

Figure 6 – Résultats expérimentaux.

requête sur l'ensemble des descriptions centralisées. Ce résultat ne fait que confirmer les travaux décrits dans (Pérez *et al.*, 2006) qui démontre que la complexité d'évaluation d'une requête SPARQL (pour des requêtes simples) dépend non seulement de la taille des descriptions RDF mais aussi du nombre de variables de la requête.

L'évaluation de la Requête 2 montre également que répondre à une requête relaxée se fait de manière plus rapide que répondre à une requête non-relaxée. De plus, répondre à une requête relaxée sur notre résumé de métadonnées est d'autant plus efficace que répondre à la même requête sur l'ensemble des métadonnées centralisées. En ce qui concerne la Requête 3, les résultats montrent que lorsqu'une variable est fixée dans une requête relaxée (par exemple, pour chercher des serveurs) son temps d'évaluation est encore plus rapide, surtout lorsque celle-ci est évaluée sur notre résumé.

Le deuxième tableau présente l'évaluation de requêtes SPARQL contenant plus de variables, c'est-à-dire les requêtes 4, 5 et 6. Ces tests montrent l'évolution du temps de réponse en fonction de la complexité des requêtes. Il est important de souligner que le résumé de métadonnées est également efficace pour ce type de requêtes.

Pour conclure, nous avons montré que l'utilisation d'un résumé sur le serveur central dans le cadre de métadonnées réparties sur différents serveurs distants améliore les performances de réponse d'une requête.

6. Conclusion

Nous avons proposé une approche qui permet d'optimiser la recherche d'information lorsque les contenus multimédias ainsi que leurs métadonnées sont distribués. Celle-ci utilise un résumé de métadonnées qui permet d'exécuter simultanément les requêtes de l'utilisateur uniquement sur certains serveurs distants qui contiennent potentiellement l'information souhaitée. À l'aide de technologies issues du Web Sémantique, nous avons présenté une méthode pour construire un tel résumé et démontré expérimentalement son efficacité. Bien entendu, notre proposition ne se limite pas aux seules descriptions RDF mais peut être utilisée pour d'autres formalismes, tels que les Topic Maps (Biezunski *et al.*, 1999), ou d'autres standards XML de métadonnées.

Dans l'avenir, nous souhaitons évaluer l'influence d'utilisation de différents degrés de résumés sur les performances du système d'information, et surtout trouver le meilleur degré de résumé possible. Nous envisageons également de spécifier d'autres méthodes pour résumer des collections de métadonnées, notamment en attribuant des poids d'importance aux éléments présents dans les descriptions multimédias.

7. Bibliographie

Alkhateeb F., Querying RDF(S) with regular expressions, PhD thesis, Université Joseph Fourier de Grenoble, 2008.

- Ayache S., Gensel J., Quénot G. M., « CLIPS-LSR experiments at TRECVID 2006 », *TRECVID'2006 Workshop*, 2006.
- Berners-Lee T., Hendler J., Lassila O., « The Semantic Web », *Scientific American*, vol. 284, n° 5, p. 34–43, May, 2001.
- Biezunski M., Bryan M., Newcomb S. R., Topic Maps : Information Technology – Document Description and Markup Languages, Specification, ISO/IEC 13250 :2000, December, 1999.
- Gauvain J. L., Lamel L., Adda G., « Audio Partitioning and Transcription for Broadcast Data Indexation », *Multimedia Tools and Applications*, vol. 14, n° 2, p. 187–200, 2001.
- Hinds N., Ravishankar C. V., « Managing metadata for distributed information servers », *Proceedings of the Thirty-First Hawaii International Conference on System Sciences*, p. 513–522, 1998.
- Hurtado C. A., Poulouvassilis A., Wood P. T., « Query Relaxation in RDF », *Journal on Data Semantics X*, vol. 4900, p. 31–61, 2008.
- Kankanhalli M. S., Chua T.-S., « Video modeling using strata-based annotation », *IEEE Multimedia*, vol. 7, n° 1, p. 68–74, 2000.
- Lambolez P., Queille J., Chrisment C., « EXREP : A Generic Rewriting Tool for Textual Information Extraction », *Ingénierie des Systèmes d'Information*, vol. 3, p. 471–487, 1995.
- Ma W. Y., Manjunath B. S., « NeTra : a toolbox for navigating large image databases », *Multimedia Systems*, vol. 7, n° 3, p. 184–198, 1999.
- Manola F., Miller E., RDF Primer, Recommendation, W3C, 2004.
- Mostefaoui G. K., Pasquier-Rocha J., Brezillon P., « Context-Aware Computing : A Guide for the Pervasive Computing Community », *Proceedings of the IEEE/ACS International Conference on Pervasive Services*, p. 39–48, 2004.
- Pérez J., Arenas M., Gutierrez C., « Semantics and Complexity of SPARQL », *Proceedings of the 5th International Semantic Web Conference*, p. 30–43, 2006.
- Pietarila P., Westermann U., Järvinen S., Korva J., Lahti J., Löthman H., « CANDELA – Storage, Analysis, and Retrieval of Video Content in Distributed Systems », *Proceedings of the IEEE Int. Conf. on Multimedia and Expo (ICME)*, p. 1557–1560, 2005.
- Prud'hommeaux E., Seaborne A., SPARQL Query Language for RDF, Recommendation, W3C, January, 2008.
- Quilitz B., Leser U., « Querying Distributed RDF Data Sources with SPARQL », *In Proceedings of 5th European Semantic Web Conference*, p. 524–538, 2008.
- Roth V., Peters J., Pinsdorf U., « A distributed content-based search engine based on mobile code and web service technology », *Scalable Computing : Practice and Experience*, vol. 7, n° 4, p. 101–117, 2006.
- Strang T., Linnhoff-popien C., Frank K., « CoOL : A Context Ontology Language to enable Contextual Interoperability », *LNCS 2893 : Proceedings of 4th IFIP WG 6.1 International Conference on Distributed Applications and Interoperable Systems*, p. 236–247, 2003.